Application Layer

Par Dr RIAHLA Mohamed Amine

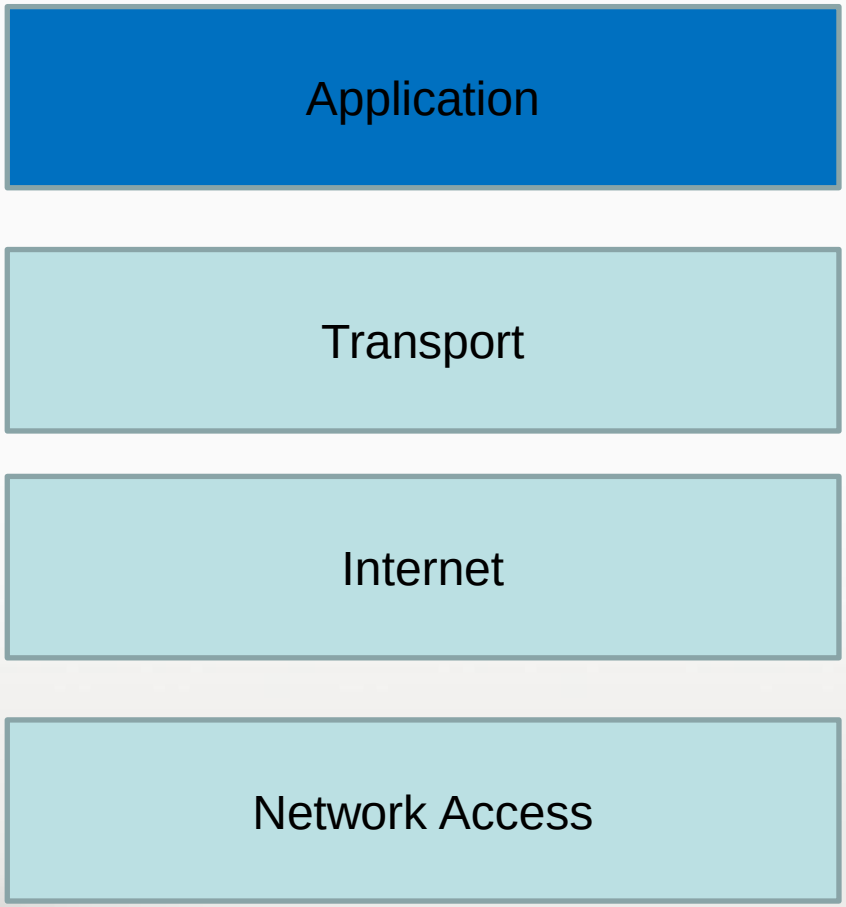
Study of TCP/IP Layers: Application Layer

- **Applications**

Browser
Messenger
Viber
...and many others!!!!!

- **Services**

Web
Mail
VoIP
...and many others!!!!!

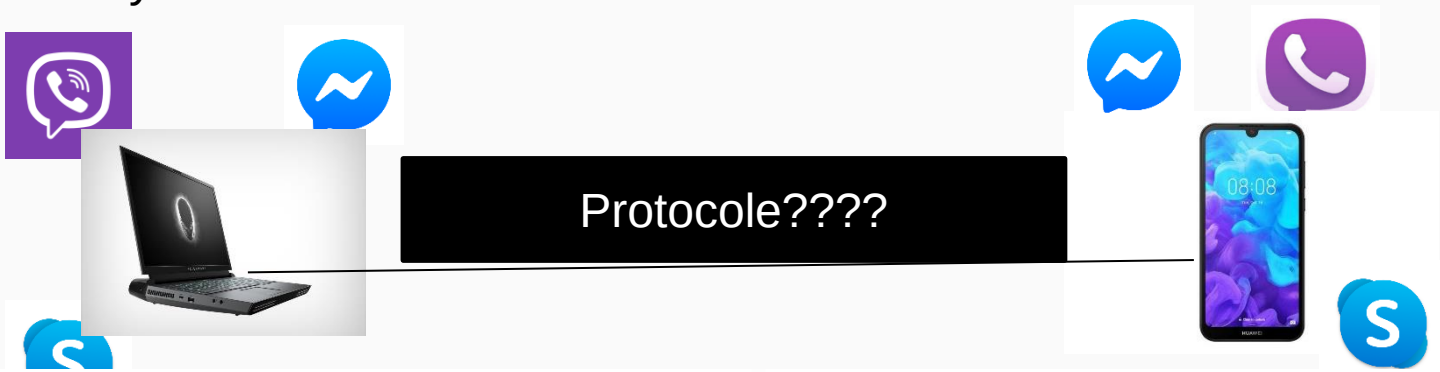


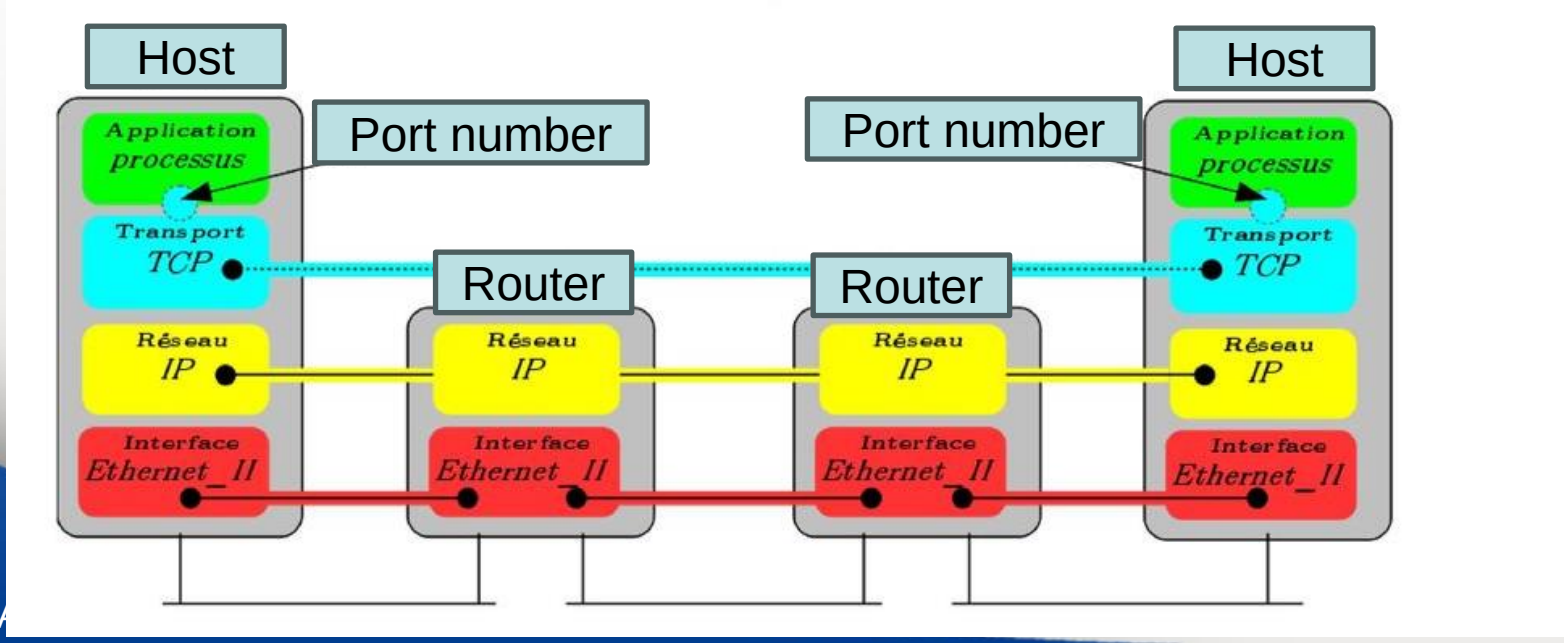
- **Protocols**

HTTP Protocol
FTP Protocol
SMTP Protocol
...and many others!!!!!

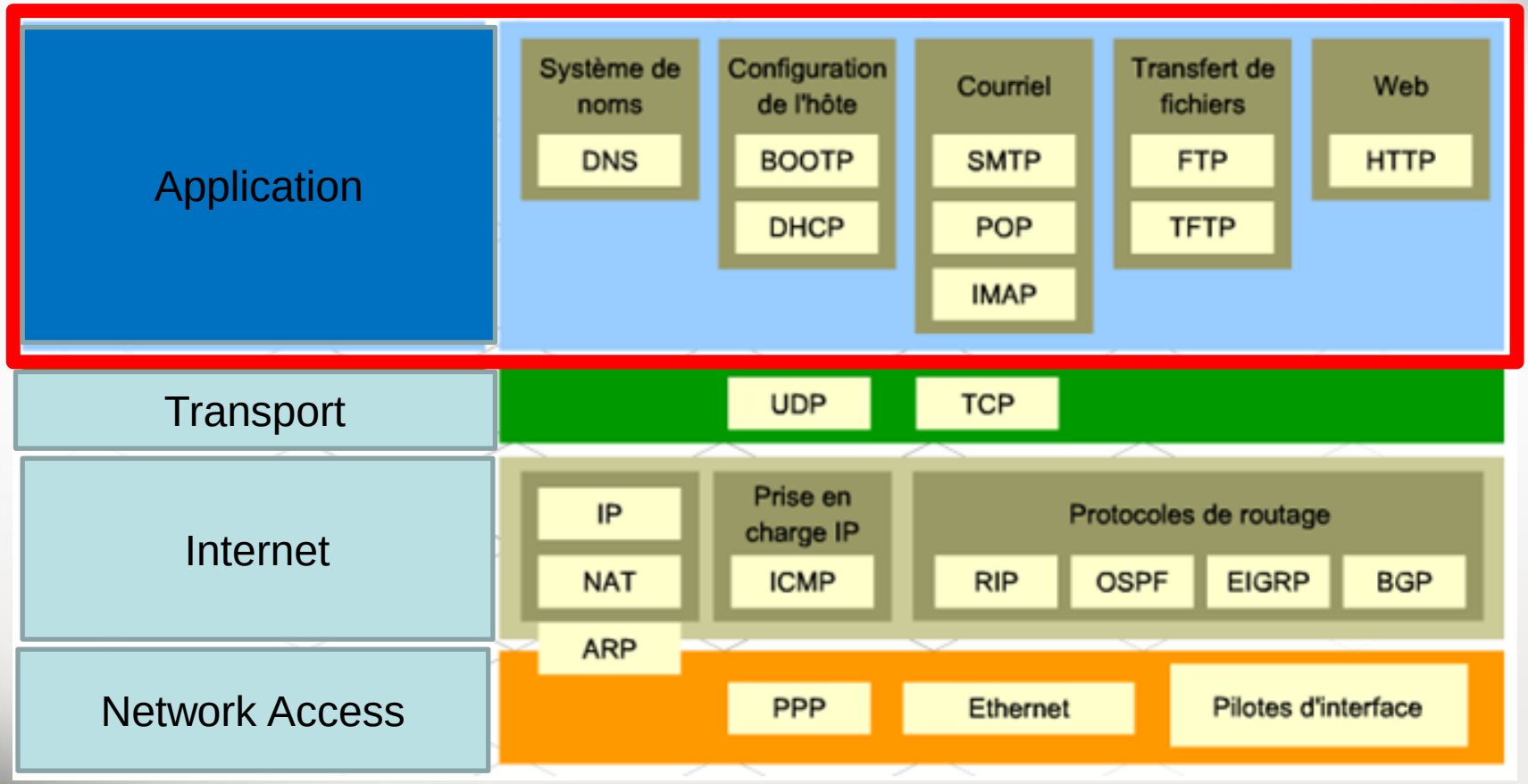
Application Layer

- It is located at the top of the TCP/IP protocol layers.
- It contains network applications that enable communication through the lower layers.

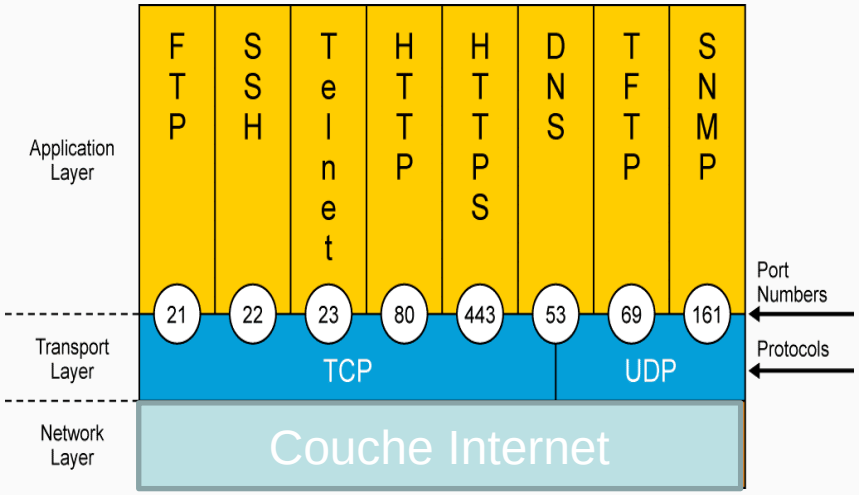




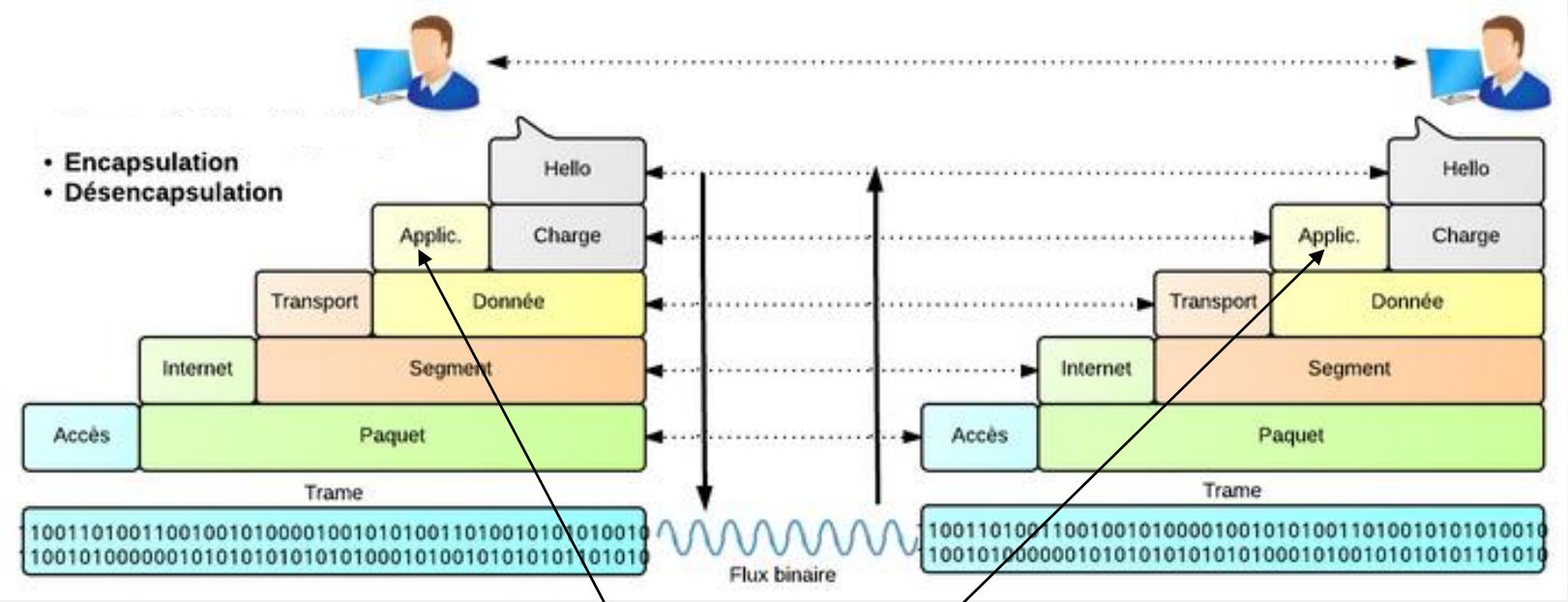
Application Layer Protocols: Client-Server



Transport Layer
Protocols either with TCP or UDP



Application Layer Headers



Each application has its own headers

Application Layer Protocol/Service/Software

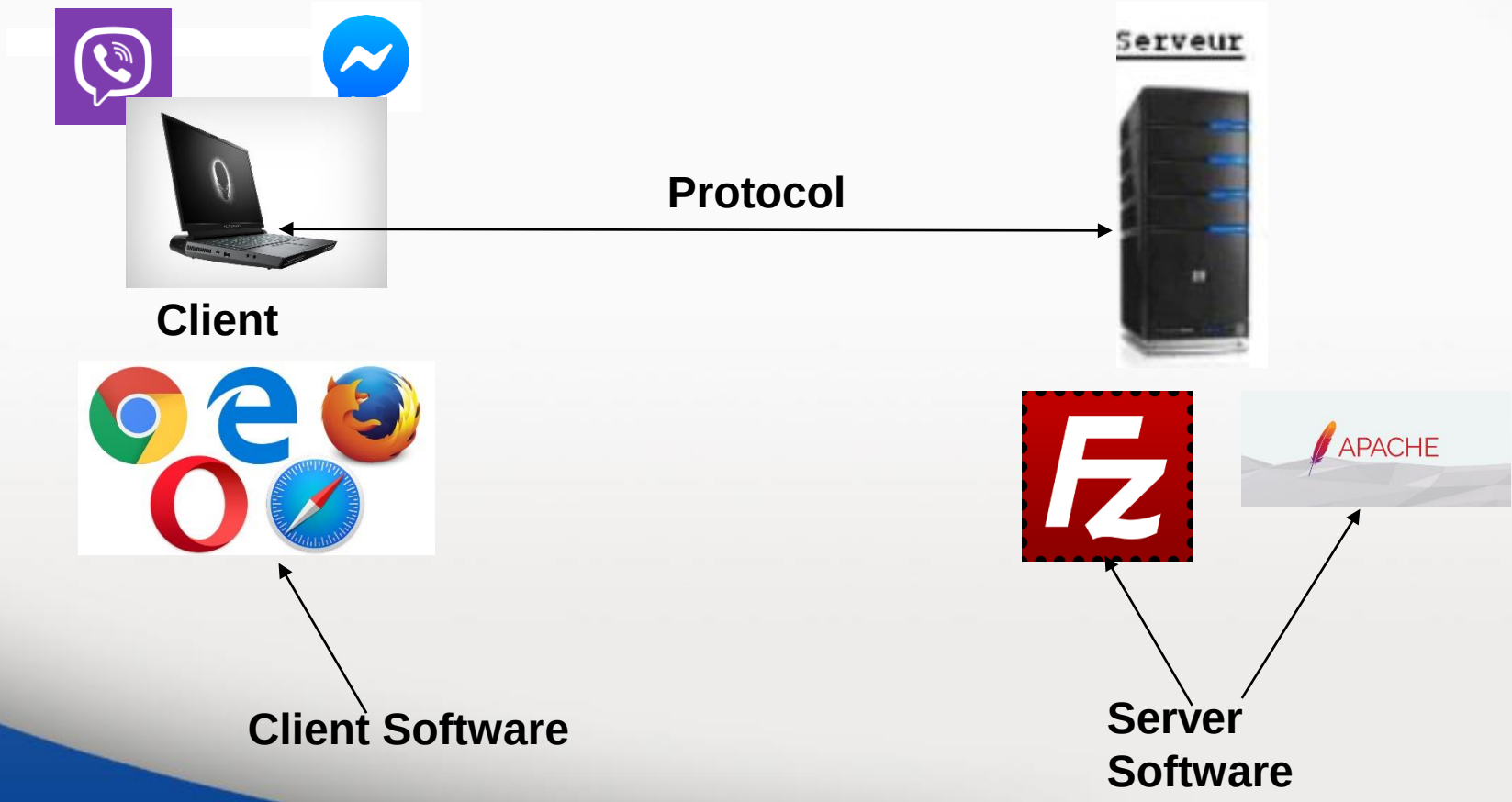
- **We say: software, process, or application.**

To provide a **service**, we use **software** (or process or application) and a transport mechanism, which is the **protocol**.

Example:

To provide the **file download service**, we use the Filezilla **application** and the FTP **protocol**.

Application Layer Protocol/Service/Software



Application Layer Services

- They can be classified according to the services they provide:
- File and print management (transfer) services,
- Remote connection services,
- Various internet utilities,
- …etc.







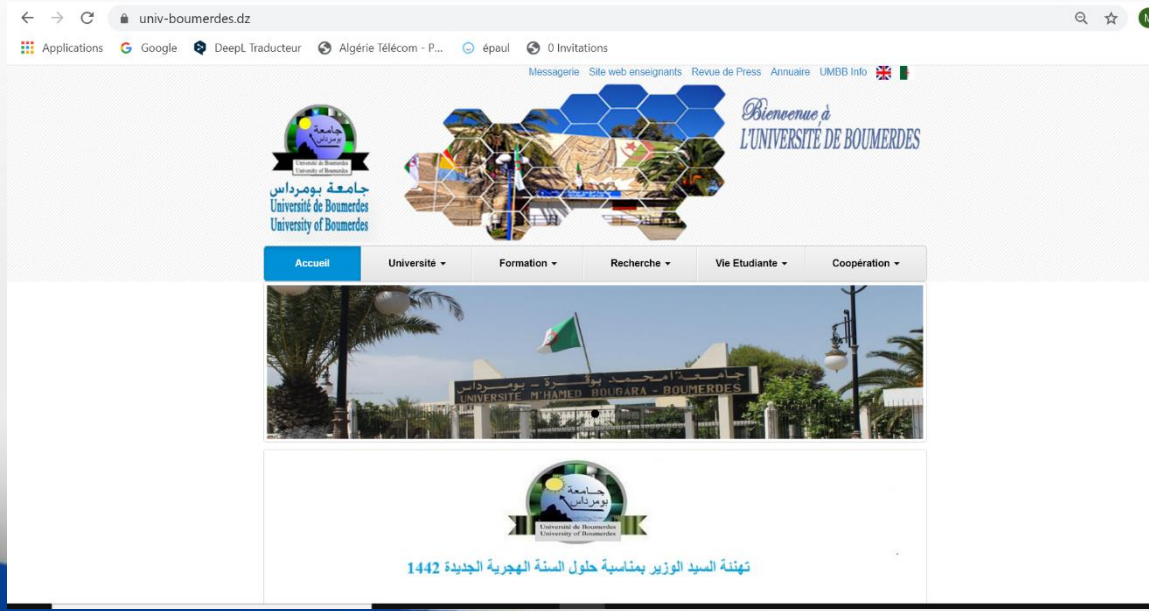
- The different protocols existing in this layer: TELNET, SSH, FTP, DNS, SNMP, HTTP, SMTP, POP...

Application Layer - HTTP

- **Port 80** with **TCP**
  Client: browser
  Server: web server

- Any web client communicates with port 80 of an HTTP server through one or more simultaneous TCP connections.

- **The web communication protocol allows the exchange of hypertext documents.**:
  - Text,
  - Static images,
  - Animated images,
  - Sound



Application Layer
- HTTP/HTML

HTTP:protocol designed to transfer text from a server to a client

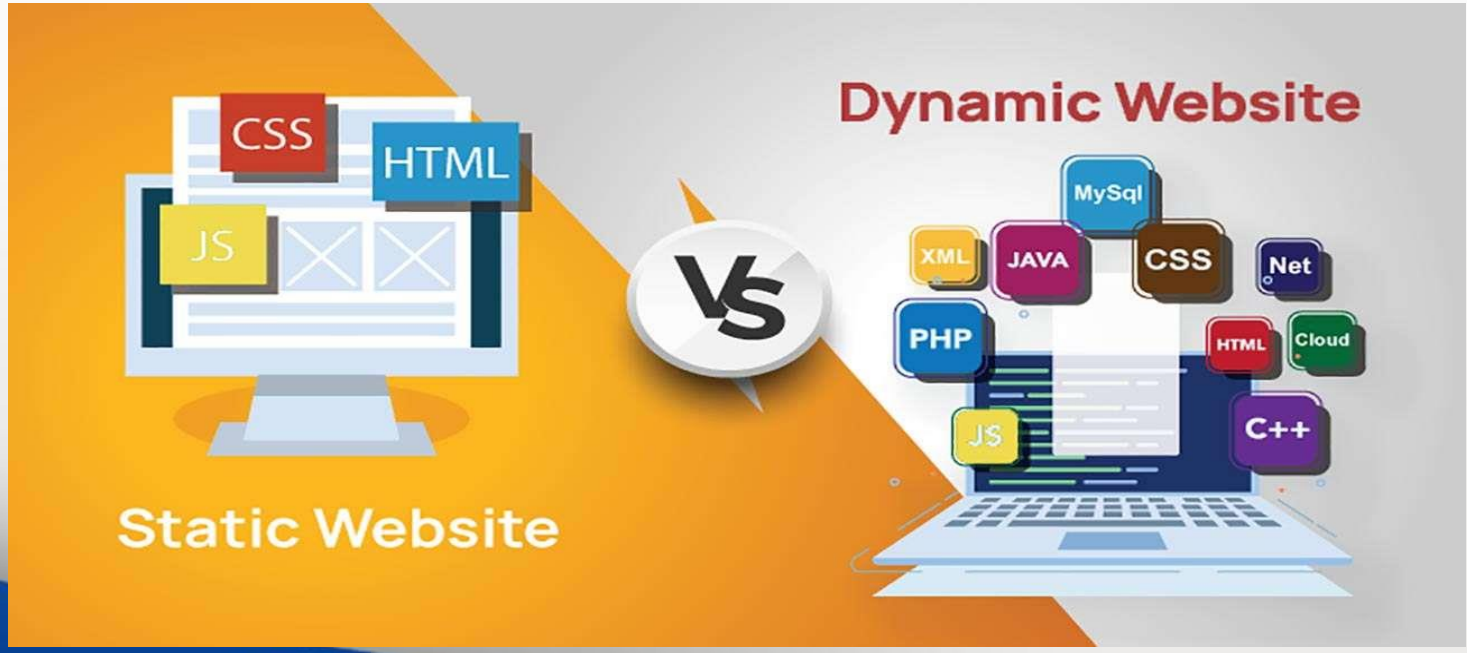
Static content vs Dynamic content

Static Page

The content of an HTML page is hard-coded in a static way, using a specialized tool or not.

Dynamic Page

The HTML is generated on the fly by the server, dynamically, based on information stored in databases or text files.



URL (Unified Ressource Locator)

http://www.debian.org/doc/manuals/reference/ch-preface.fr.html:

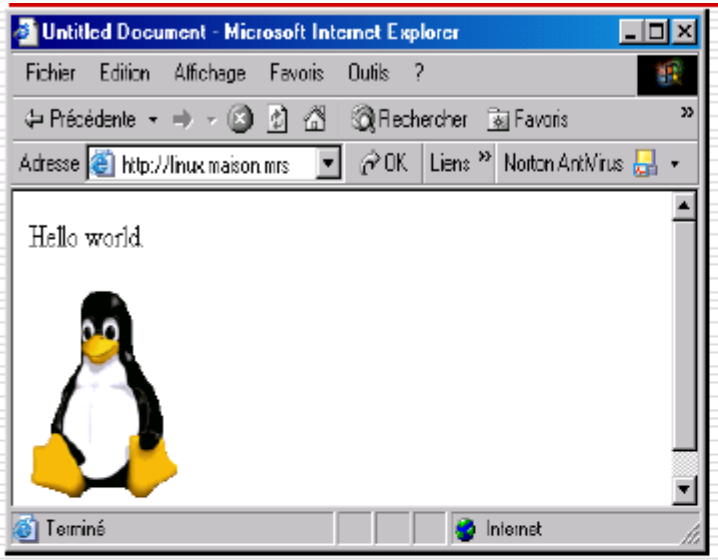
http:
The used protocol

//www.debian.org
The web server containing the requested information

/doc/manuals/reference/ch-preface.fr.html:
The full path of the visited page within the directory structure of the target server.

http://www.debian.org OR
http://www.debian.org/***index.html***.

Query an HTTP server using Telnet




```
<html>
<head>
    <title>Untitled Document</title>
    <meta http-equiv= "Content-
Type"                content=
"text/html; charset=iso-8859-1">
</head>

<body bgcolor= »#FFFFFF » text=
»#000000 »>
 <p>Hello world.</p>
<p><img src="images/tux.gif"
width= "97" height= "115"></p>
</body>
</html>
```

Query an HTTP server using Telnet

telnet 127,0,0,1 80

GET / HTTP/1.0

HTTP/1.1 200 OK
Date: Wed, 08 May 2002 14:26:57 GMT
Server: Apache-AdvancedExtranetServer/1.3.23 (Mandrake Linux/4mdk) auth_ldap/1.6.0 mod_ssl/2.8.7 OpenSSL/0.9.6c PHP/4.1.2
Last-Modified: Sun, 14 Apr 2002 09:29:32 GMT
ETag: "57d44-116-3cb94bfc"
Accept-Ranges: bytes
Content-Length: 278
Connection: close
Content-Type: text/html

HTTP Request

GET /~ferment/http/prog/page_test1.html HTTP/1.1
Connection: Keep-Alive
User-Agent: Mozilla/5.0 (compatible; Konqueror/3.1; Linux; Fr)
Referer: http://www.u-picardie.fr/~ferment/http/prog/
Pragma: no-cache
Cache-control: no-cache
Accept: text/html, image/jpeg, image/png, text/*, image/*, */*
Accept-Encoding: x-gzip, x-deflate, gzip, deflate, identity
Accept-Charset: iso-8859-1, utf-8; q=0.5, *; q=0.5
Accept-Language: fr, en
Host: www.u-picardie.fr

.....

HTTP Replay

HTTP/1.1 200 OK
Date: Tue, 22 Jun 2004 13:18:15 GMT
Server: Apache/1.3.26 (Unix) Debian GNU/Linux PHP/4.1.2 mod_ssl/2.8.9 OpenSSL/0.9.6g DAV/1.0.3
Last-Modified: Tue, 22 Jun 2004 13:15:43 GMT
ETag: "63f3d-8e-40d830ff
Accept-Ranges: bytes
Content-Length: 142
Keep-Alive: timeout=15, max=2000
Connection: Keep-Alive
Content-Type: text/html
<Html> <Body><h1>page html **</h1> ** contenant une image **
et une seule<img src="balle.gif">**
** **
</Body><Html>

HTTP Replay

Code	Class	Usage
1xx	Informational	The request has been received and the process is continuing...
2xx	Success	The action was successfully received, understood, and accepted...
3xx	Redirection	Further action must be taken to complete the request...
4xx	Client Error	The request contains bad syntax or cannot be fulfilled...
5xx	Server Error	The request is valid, but the server failed to fulfill it...

HTTP Replay

Status Code	Action	Description
200	OK	Successfully retrieved resource
201	Created	A new resource was created
204	No Content	Request has nothing to return
301 / 302	Moved	Moved to another location (redirect)
400	Bad Request	Invalid request / syntax error
401 / 403	Unauthorized	Authentication failed / Access denied
404	Not Found	Invalid resource was requested
409	Conflict	Conflict was detected, e.g. duplicated email
500 / 503	Server Error	Internal server error / Service unavailable

HTTP and TCP protocols

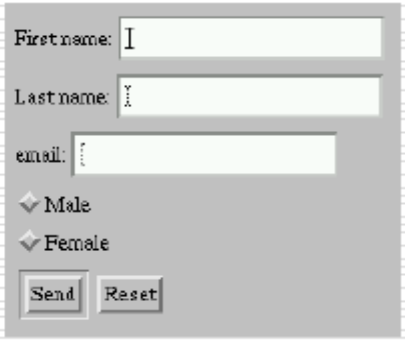
No.	Time	Source	Destination	Proto Info
1	0.000000	192.168.0.10	192.168.0.253	TCP 1282 > 80 [SYN]
2	0.000163	192.168.0.253	192.168.0.10	TCP 80 > 1282 [SYN, ACK]
3	0.000565	192.168.0.10	192.168.0.253	TCP 1282 > 80 [ACK]
4	0.001410	192.168.0.10	192.168.0.253	HTTP GET / HTTP/1.1
5	0.001487	192.168.0.253	192.168.0.10	TCP 80 > 1282 [ACK]
6	0.068550	192.168.0.253	192.168.0.10	HTTP HTTP/1.1 200 OK
7	0.098435	192.168.0.10	192.168.0.253	HTTP GET /images/tux.gif HTTP/1.1
8	0.098593	192.168.0.253	192.168.0.10	TCP 80 > 1282 [ACK]
9	0.099450	192.168.0.253	192.168.0.10	HTTP HTTP/1.1 200 OK
10	0.099724	192.168.0.253	192.168.0.10	HTTP Continuation
11	0.102794	192.168.0.10	192.168.0.253	TCP 1282 > 80 [ACK]
12	0.102915	192.168.0.253	192.168.0.10	HTTP Continuation
13	0.280331	192.168.0.10	192.168.0.253	TCP 1282 > 80 [ACK]

Cache request

No.	Time	Source	Destination	Proto	Info
1	0.000000	192.168.0.10	192.168.0.253	TCP	2632 > 80 [SYN]
2	0.000144	192.168.0.253	192.168.0.10	TCP	80 > 2632 [SYN, ACK]
3	0.000540	192.168.0.10	192.168.0.253	TCP	2632 > 80 [ACK]
4	0.001342	192.168.0.10	192.168.0.253	HTTP	GET / HTTP/1.1
5	0.001461	192.168.0.253	192.168.0.10	TCP	80 > 2632 [ACK]
6	0.004186	192.168.0.253	192.168.0.10	HTTP	HTTP/1.1 304 Not Modified
7	0.200392	192.168.0.10	192.168.0.253	TCP	2632 > 80 [ACK]

Forms

```
<FORM action="http://somesite.com/prog/adduser"
method="post">
<P>
First name:
<INPUT type="text" name="firstname"><BR>
Last name:
<INPUT type="text" name="lastname"><BR>
email: <INPUT type="text" name="email"><BR>
<INPUT type="radio" name="sex" value="Male">
Male<BR>
<INPUT type="radio" name="sex" value="Female">
Female<BR> <INPUT type="submit" value="Send">
<INPUT type="reset">
</P>
</FORM>
```



And many other things about HTTP

- **HTTPMethods:**GET, HEAD, POST, OPTIONS, CONNECT, TRACE, PUT, PATCH, DELETE
- **Cookies**

```
Frame 20 (1473 on wire, 1473 captured)
...
Hypertext Transfer Protocol
    HTTP/1.1 200 OK\r\n
    Date: Sun, 21 Apr 2002 13:11:11 GMT\r\n
    Server: Apache-AdvancedExtranetServer/1.3.22
            (Mandrake Linux/1.1mdk) PHP/4.0.6 mod_ssl/2.8.5 OpenSSL/0.9.6b\r\n
    X-Powered-By: PHP/4.0.6\r\n
    Set-Cookie: VotreCookie=cookie; expires=Sun, 21-Apr-02 13:13:11 GMT\r\n
    Le cookie a un identificateur : VotreCookie et une valeur : cookie
    Keep-Alive: timeout=15, max=99\r\n
    Connection: Keep-Alive\r\n
    Transfer-Encoding: chunked\r\n
    Content-Type: text/html\r\n
    \r\n

    Data (1051 bytes)
Vient ensuite le document lui-même...
0000  34 30 66 0d 0a 3c 68 74 6d 6c 3e 0d 0a 0d 0a 3c   40f..<html>....<
0010  68 65 61 64 3e 0d 0a 3c 6d 65 74 61 20 68 74 74   head>..<meta htt
...
```

- **HTTPProxy**
- **The 'transparent' proxy:**
- **Persistent connections:** Keep-Alive Connection
- **Managing an HTTP connection with pipelining: Multiple GET**
- **Content Negotiation**
- **HTTP Authentication:**Basic, Digest

HTTP versions

Feature	HTTP/0.9	HTTP/1.0	HTTP/1.1	HTTP/2	HTTP/3
Persistent Connections	No	No	Yes	Yes	Yes
Header Compression	No	No	No	Yes	Yes
Multiplexing	No	No	No	Yes	Yes
Binary (instead of text)	No	No	No	Yes	Yes
Secure by Default	No	No	No	No	Yes (TLS 1.3)
Uses UDP Protocol	No	No	No	No	Yes
Examples of Usage	Very limited	Basic web servers	Dynamic web and apps	Modern web, apps, CDN	Low-latency connections (e.g., mobile, streaming)

The versions of HTTP have evolved to address growing needs for performance, security, and connection management. HTTP/1.1 is still widely used, but newer versions like HTTP/2 and HTTP/3 offer notable improvements in speed, security, and connection handling. HTTP/3, based on QUIC, is especially useful in mobile environments and high-latency networks.

Tools

Windows

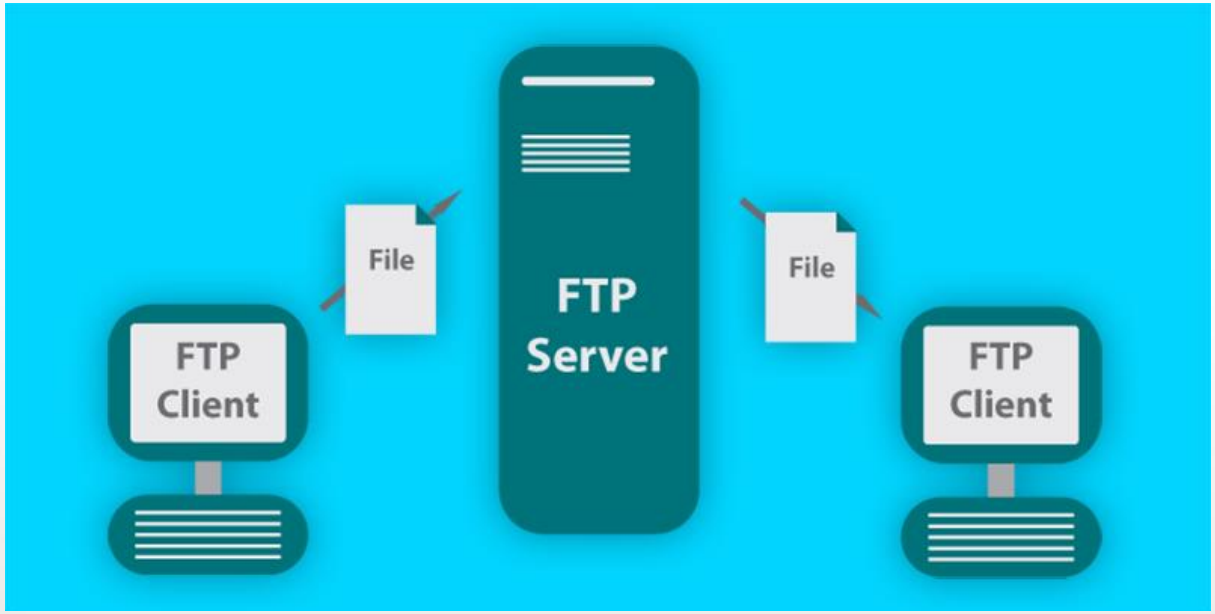
- **Wamp, easyphp,,,etc**

Linux

- **Httpd, apache2**

Application Layer
FTP

- **Port 21 avec TCP**
- **Client:** Filezila
- **Serveur:** FTPserver

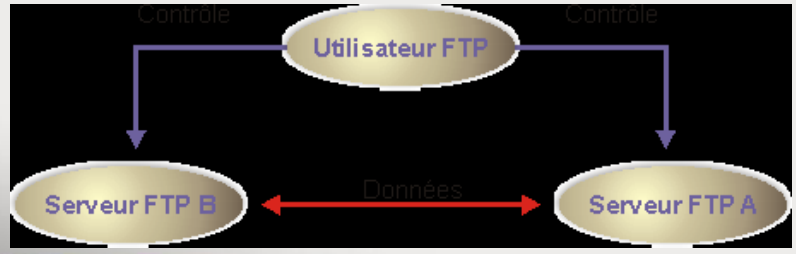


Application Layer
FTP

- The File Transfer Protocol, or FTP, is a communication protocol designed for the transfer of files over a TCP/IP network.
- It allows a computer to copy files to another computer on the network, upload content to a website, or even delete or modify files on that remote computer.
- The protocol belongs to the session layer of the OSI model, and to the application layer of the ARPA model, and it uses a TCP connection.
  - There are two main secure alternatives to traditional FTP, which does **not** encrypt data by default::
    - **SFTP (SSH File Transfer Protocol)**
    - **FTPS (FTP Secure / FTP-SSL)**

Application Layer
FTP

- The client opens an FTP session on a server.
- There are many public FTP servers available. An FTP server requires client authentication.
- There is often an "**anonymous**" account, which provides read-only access to the public section of the server.
- There are also private sections where clients with an account can have write access to certain directories.
- This is the case, for example, when updating personal web pages.
    - Unlike HTTP, FTP uses at least two separate channels
        - One for exchanging protocol commands,
        - The other for transferring the actual data.

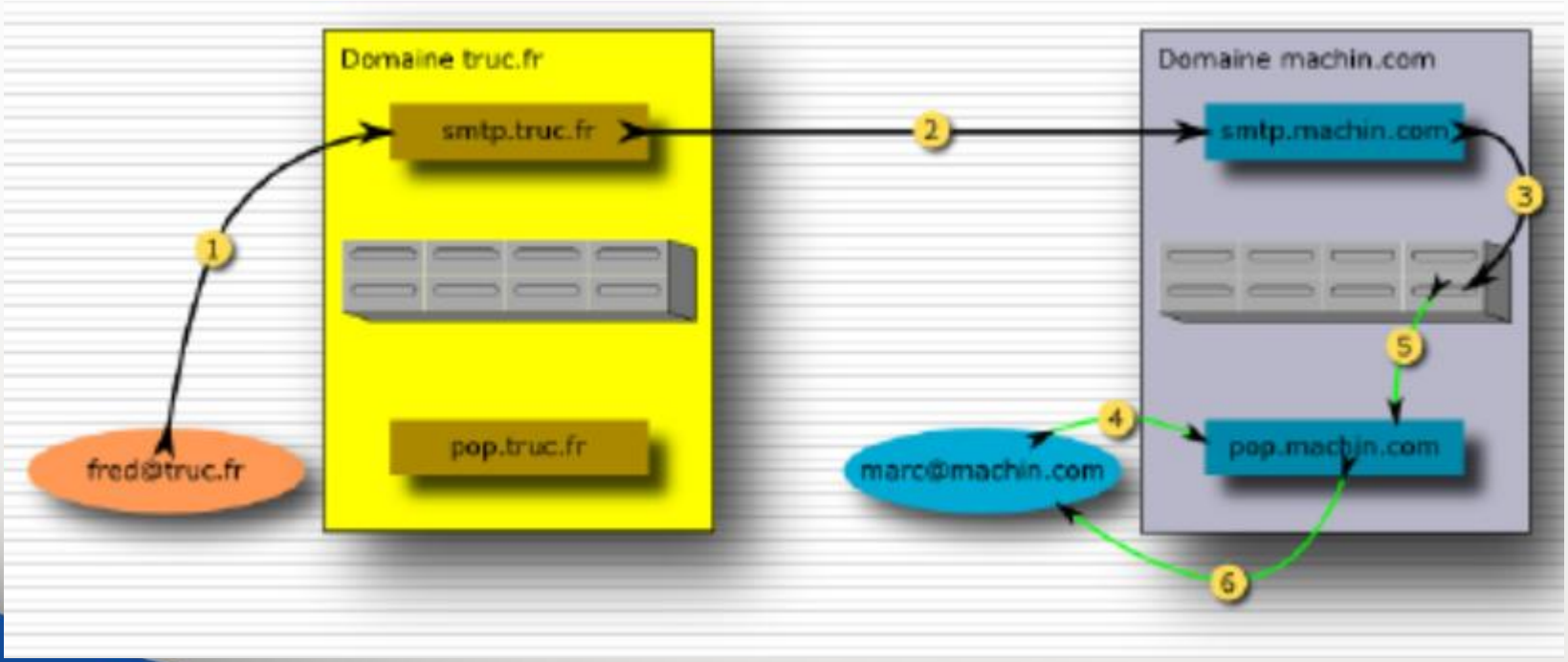


Active and Passive Modes

- **Active Mode**: The FTP client determines the connection port to be used for data transfer.
- **Passive Mode**: The FTP server itself determines the connection port to be used for the data transfer (data connection) and communicates it to the client.

Application Layer
Mail Service

- **SMTP: Port 21 avec TCP**
- **POP: Port 109/110 avec TCP**
- **IMAP: Port 143 avec TCP**



SMTP
Simple Mail Transfert Protocol

One of the most fundamental Internet protocols

- Allows messages to be transported over the Internet.
- It knows how to route a message to a mailbox, but goes no further.
- To do this, it first analyzes the part of the address to the right of the @ to find the recipient’s domain.
- If that domain is local, it then looks for the recipient's mailbox by examining the part of the address to the left of the @.
- If the recipient’s domain is not local, it looks for the SMTP server that handles that domain using the MX fields from the recipient domain’s DNS, and forwards the message to that server.

POP3 (port 110)
Post Office Protocol

Allows the user to retrieve their mail from a host that does not store their mailbox.

- Establish a TCP connection between the client and the server.
- The POP3 server is capable of responding to a number of commands.

IMAP (port)
Internet Mail Access Protocol

A protocol for retrieving email messages, functionally similar to POP.

- It is defined by RFC 2060, which was later replaced by RFC 3501.
- An IMAP server listens by default on port 143.

Application Layer
DHCP

- **Port 67** with **UDP**
- **Client:** integrated into the OS
- **Server:** DHCP server

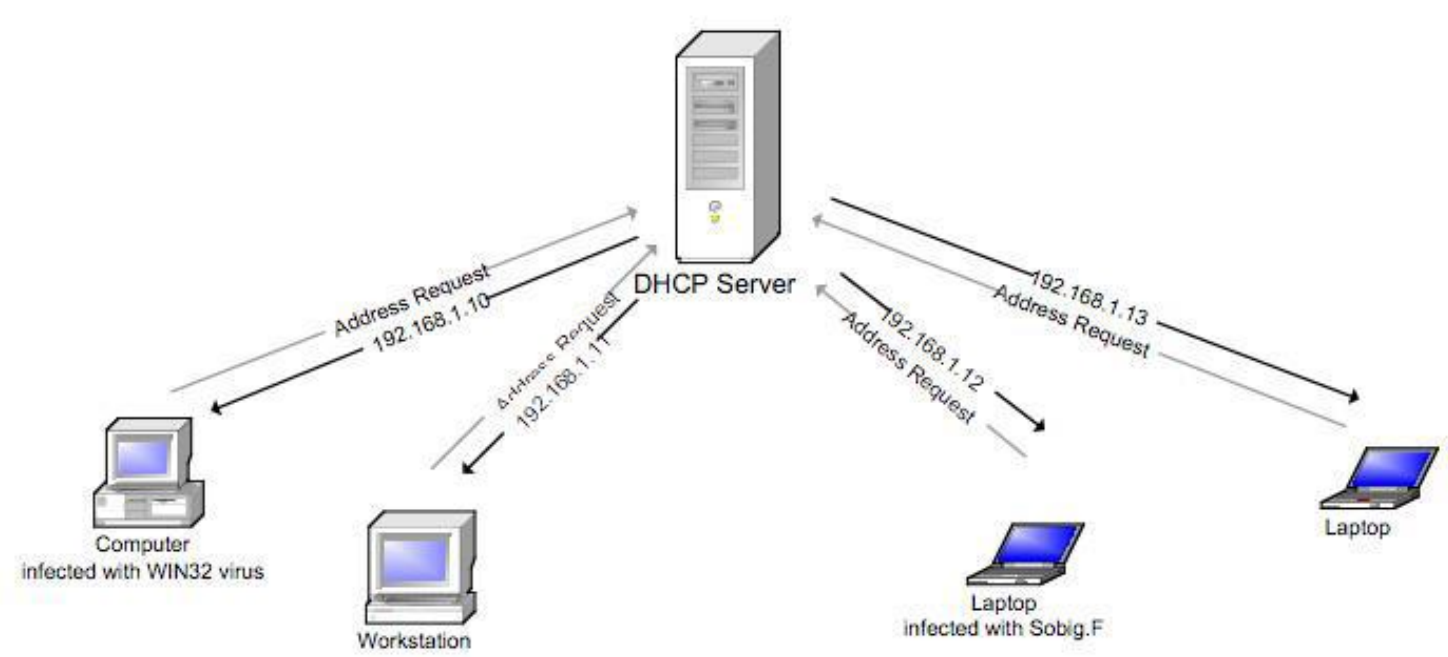


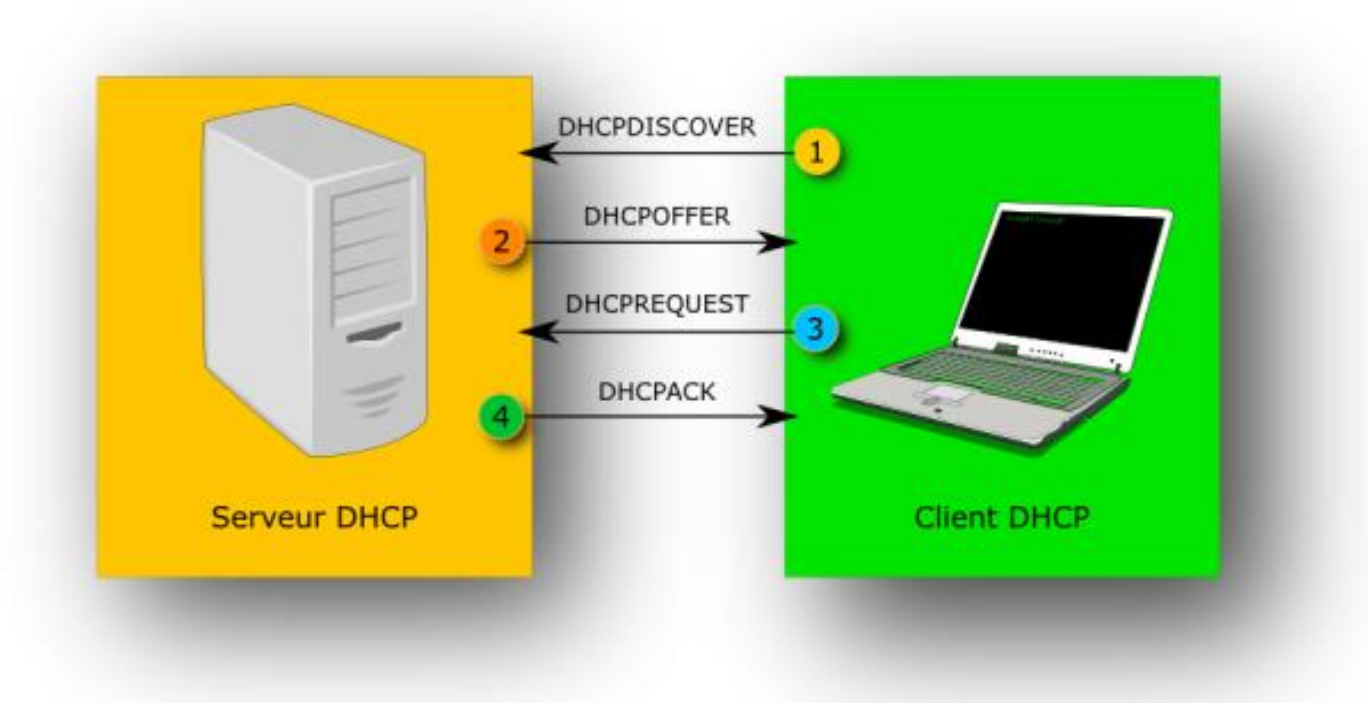
Figure 1: Standard DHCP provides IP addresses to all machines

Application Layer
DHCP

Dynamic Host Configuration Protocol (DHCP) is a network protocol whose role is to automatically configure the IP settings of a station, notably by automatically assigning it an IP address and a subnet mask.

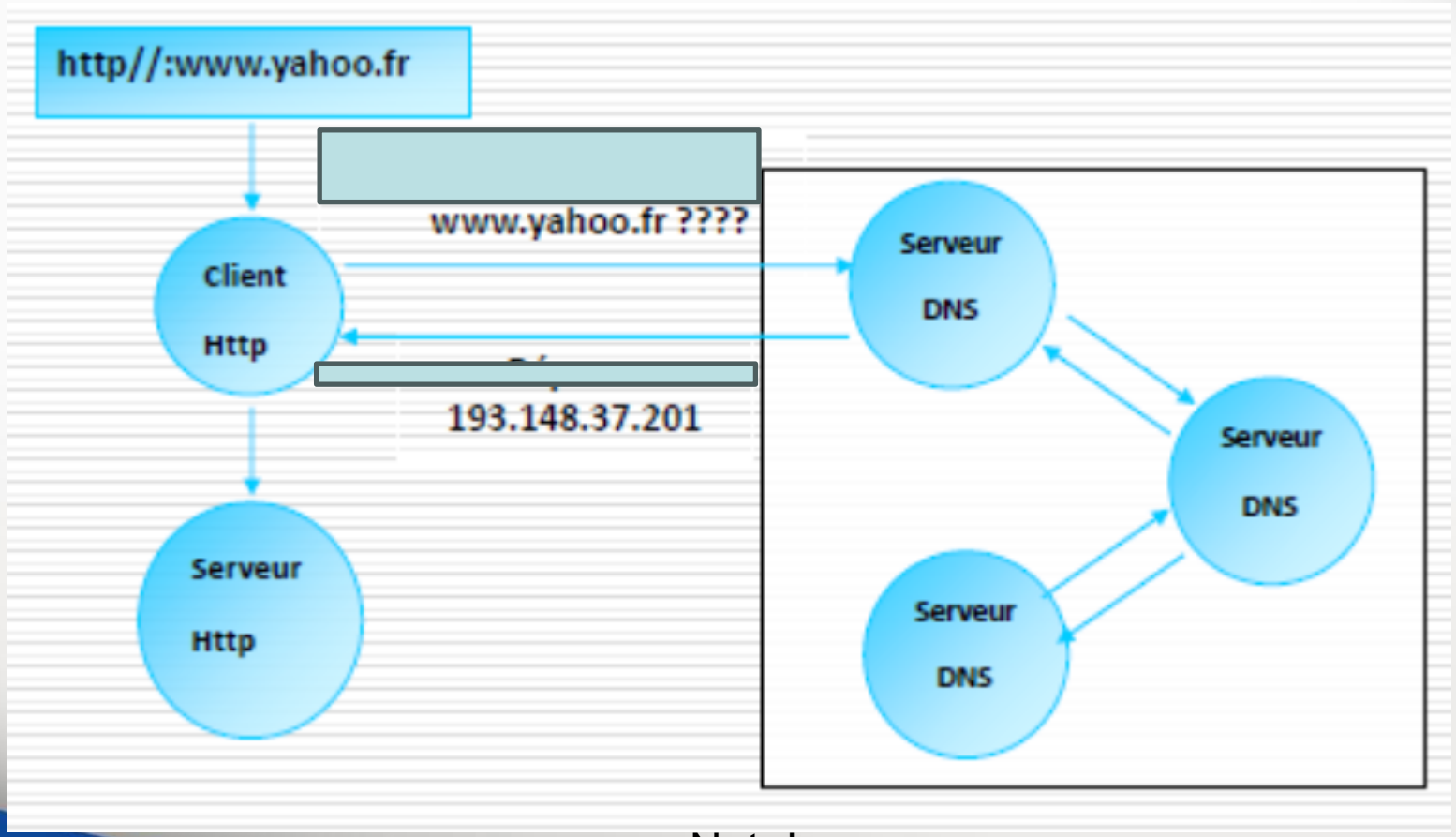
DHCP can also configure the address of the default gateway, DNS name servers, and NBNS name servers (known as WINS servers on Microsoft networks).

Application Layer
DHCP



Application Layer - DNS

- **Port 53 with UDP**
- **Client: integrated into the OS**
- **Server: DNS server**



.com, .org. Net.dz,…

Application Layer
- SNMP

A “simple” protocol designed to manage IT equipment, whether remotely or locally.

•Allows viewing of a potentially large amount of information regarding hardware, network connections, and their load status.

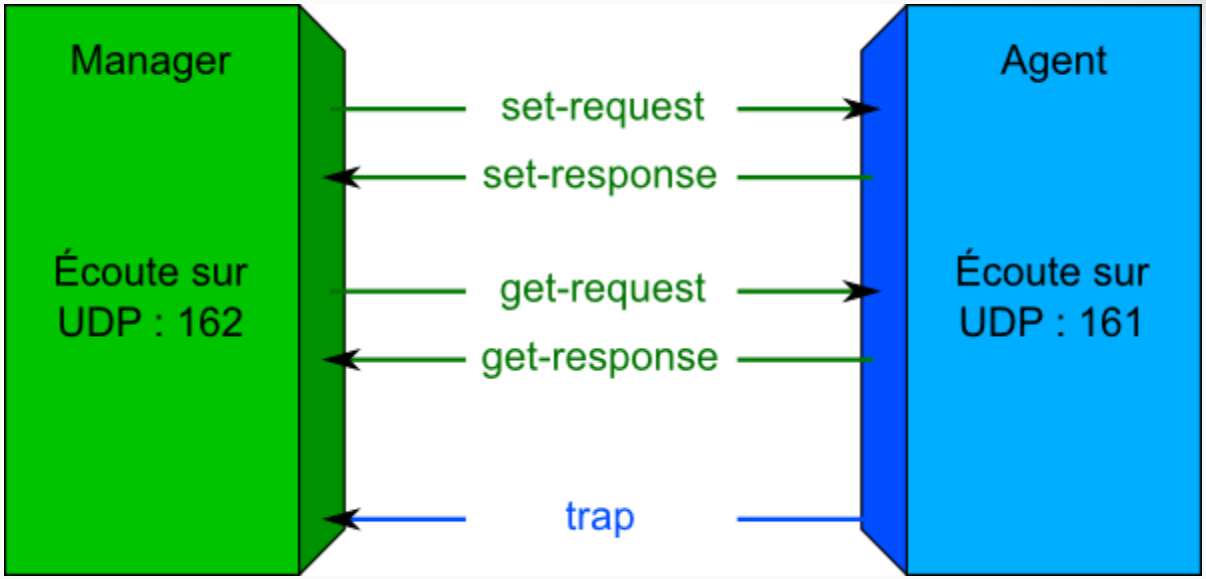
•Enables modification of certain component settings.

•Alerts the administrator in case of events considered critical.

•An administrator can have full control over the entire network and IT infrastructure without leaving their air-conditioned office.

•Some “black box” devices come with an SNMP agent and do not allow the installation of other agents, such as switches, Wi-Fi access points, and other routers.

Application Layer - SNMP



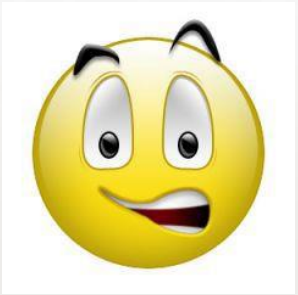
Command	Action
get-request	The SNMP Manager requests information from an SNMP Agent
get-next-request	The SNMP Manager requests the next piece of information from the SNMP Agent
set-request	The SNMP Manager updates information on an SNMP Agent
get-response	The SNMP Agent replies to a get-request or a set-request
trap	The SNMP Agent sends an alert to the Manager

Application Layer
Remote administration

- **Telnet: Port 23 avec TCP**
- **SSH: Port 22 avec TCP**
- **VPN**

Application Layer
- Other Protocols-

- **SNMP (Port 161)**
- **HTTPS (Port 443)**
- **IMAPS (Port 993)**
- **TFTP (Port 69)**
- **NTP**
- **SMB**
- **Proposez d'autres !!!!**



Exercice!!!!!